ISTITUTO ITALIANO
DI TECNOLOGIA

Intrinsically Motivated Hierarchical Learning

Stephen Hart
Robotics, Brain & Cognitive Sciences
Italian Institute of Technology
Machine Learning Day
June 8, 2010

Overview

- A grounded representation for state & action
  - A formalism in which the robot can explore its sensory and motors combinatorics using machine learning algorithms
- An intrinsic motivation function for discovering *behavioral affordances*
  - Supports autonomous skill acquisition in developmental learning stages
- Hierarchical behavior organization
  - Allows for efficient generalization and re-use
- A categorical description of the world in terms of affordances
  - Supports for long-term exploration

The Control Basis

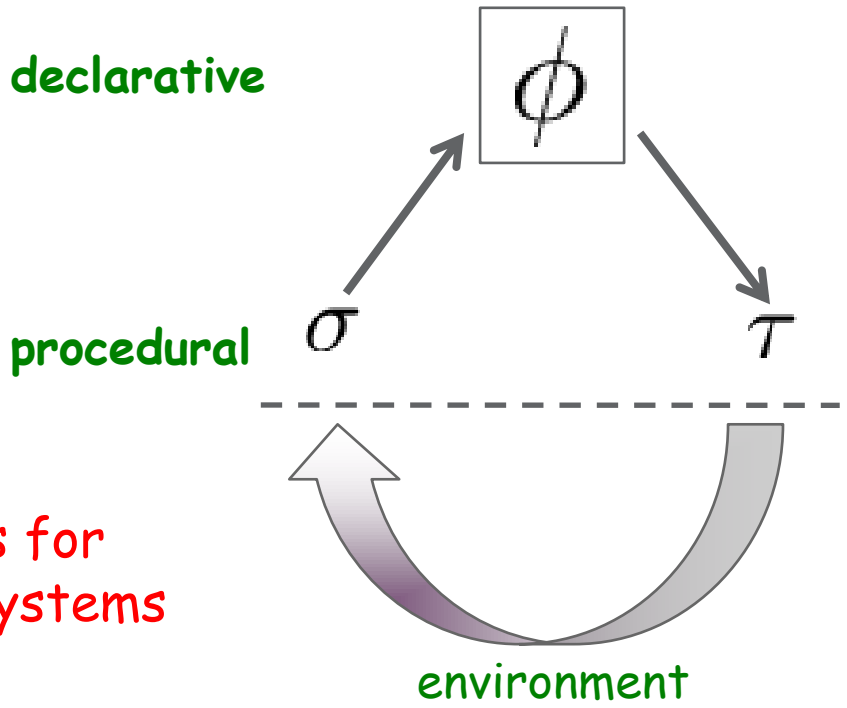
objectives with I/O constraints:

ϕ : *navigation functions*

typed resources:

σ : sensory feedback signals

τ : motor variables



A combinatorial basis for dynamical closed-loop systems

controller construction:

$$\mathbf{J} = \frac{\delta\phi(\sigma)}{\delta\tau} \qquad \Delta\tau = -\mathbf{J}^{\#}\phi(\sigma)$$

co-articulation:

$$c_2 \triangleleft c_1$$

(Huber & Grupen – IJCAI 1997)

The Control Basis

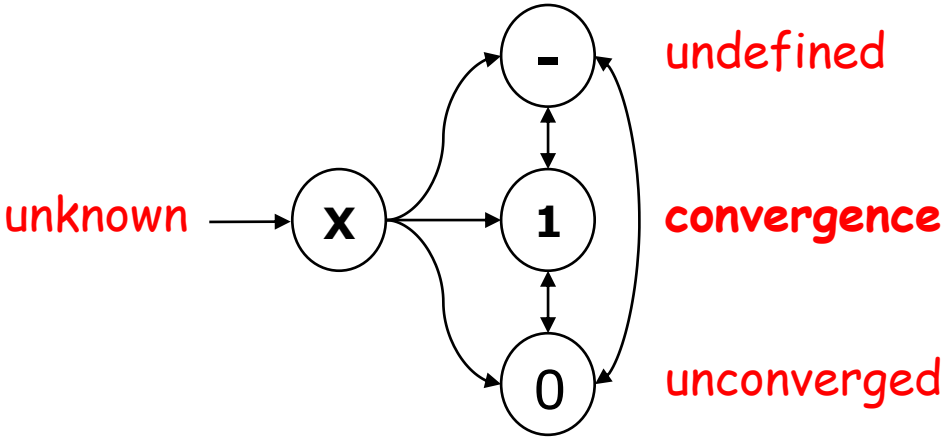
objectives with I/O constraints:

ϕ : *navigation functions*

typed resources:

σ : sensory feedback signals

τ : motor variables



A discrete state abstraction p for continuous dynamical systems

controller construction:

$$\mathbf{J} = \frac{\delta\phi(\sigma)}{\delta\tau} \qquad \Delta\tau = -\mathbf{J}^{\#}\phi(\sigma)$$

co-articulation:

$$c_2 \triangleleft c_1$$

(Hart et al. – ICRA 2008)

The Control Basis

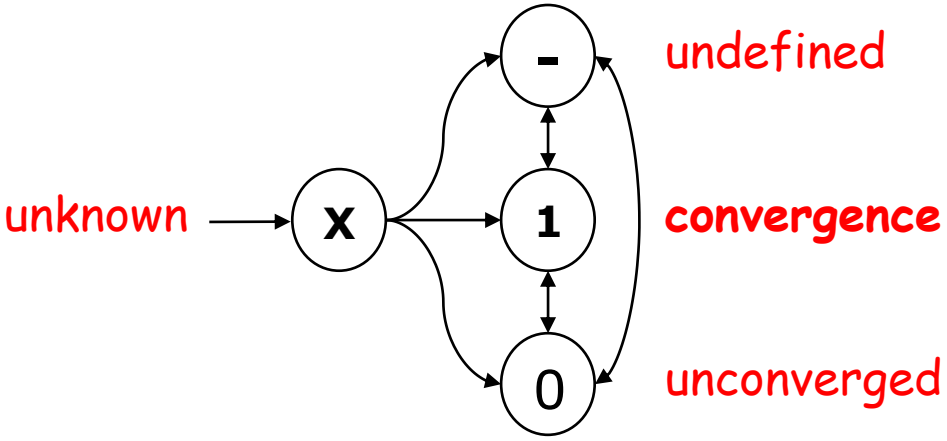
objectives with I/O constraints:

ϕ : *navigation functions*

typed resources:

σ : sensory feedback signals

τ : motor variables



A discrete state abstraction p for continuous dynamical systems

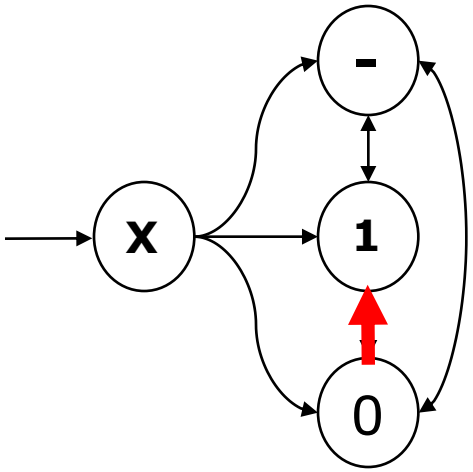
Value functions represent the *hierarchical* abstraction of low-level objectives:

the same 4-predicate logic can be used to represent the state of entire programs

(Hart et al. – ICRA 2008)

Intrinsic Reward for Affordance Discovery

- Intrinsic reward is defined by convergence of behavior **afforded** by the environment.



$$m = ((p_{k-1} \neq 1) \wedge (p_k = 1))$$

$$r = \left(m \wedge \left(\sigma \subseteq \Omega_{\sigma(env)}\right)\right)$$

set of sensory signals derived from the environment

- domain general
- leads to behavior *and* exploration

(Hart et al. – ICRA 2008)

Developmental Programming Example

- Each program is learned in distinct developmental **stage**.
  - Each program is designed to uncover a new visual- or force-domain affordance.
- Each stage is designed by the “teacher” to eliminate extraneous sensory stimuli and motor options.
- Q-learning is used to learn policies for discovering affordances.

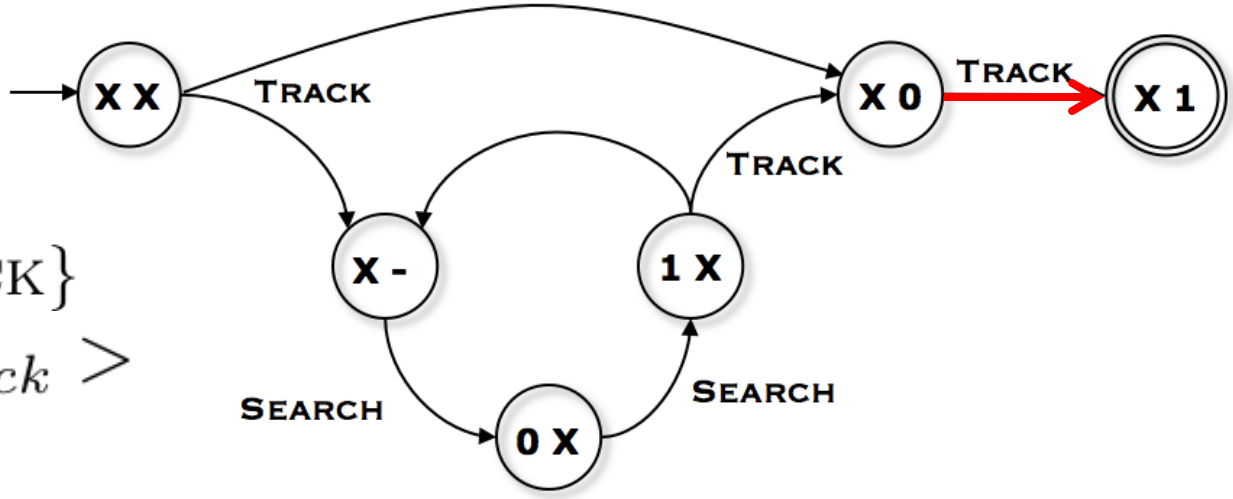
$$\Phi(s,a) \leftarrow \Phi(s,a) + \alpha\big(r + \gamma \max_{a'} \Phi(s',a') - \Phi(s,a)\big)$$

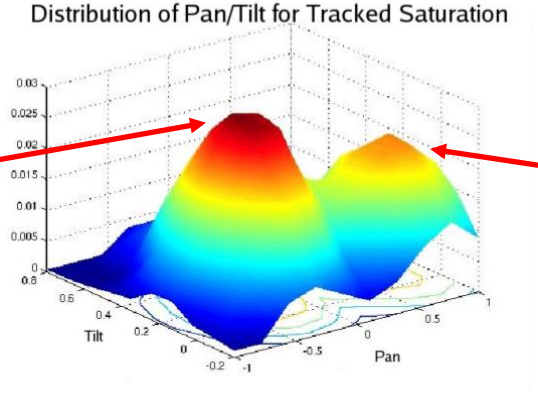
- Each training stage is limited to 25-50 learning episodes, with ϵ - greedy exploration.

STAGE 1: SearchTrack

rewardable affordance

$$\mathcal{A} = \{\text{SEARCH}, \text{TRACK}\}$$
$$\mathcal{S} = < p_{search}\ p_{track} >$$



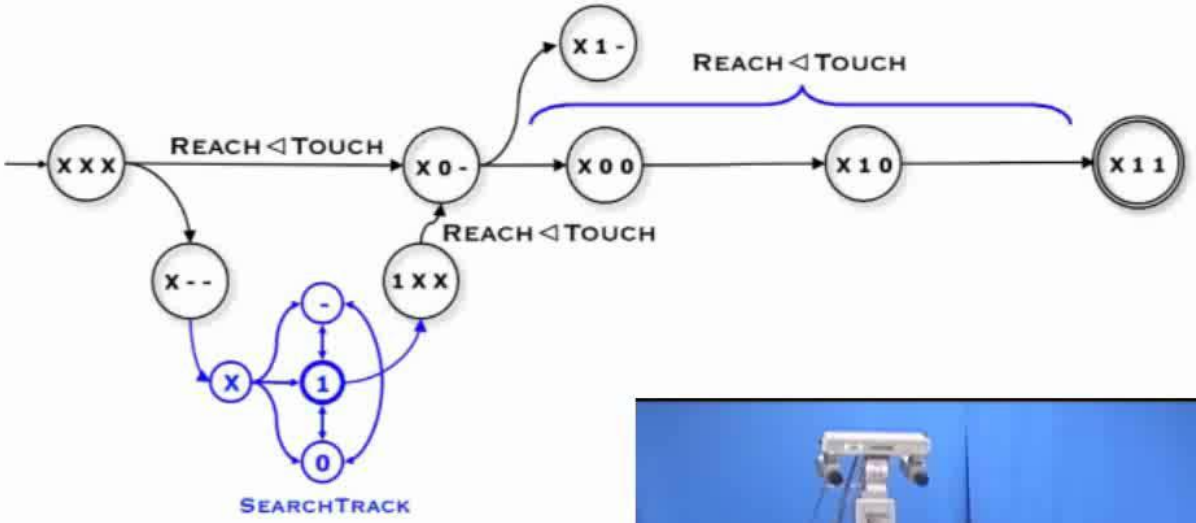


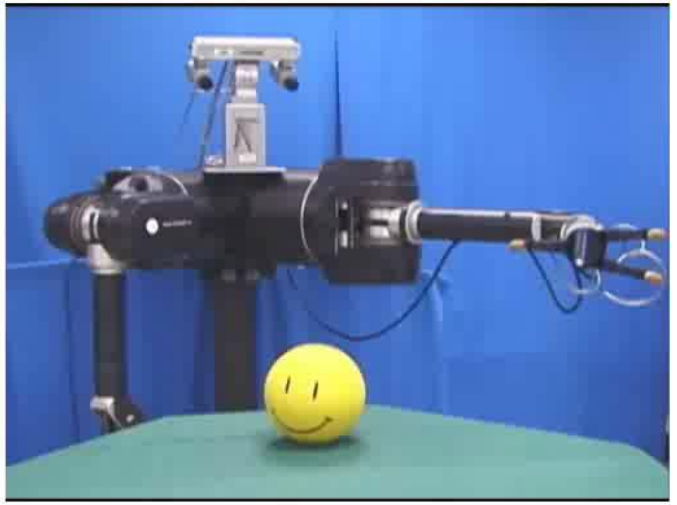
STAGE 2: ReachTouch

$$\mathcal{A} = \{\text{SEARCHTRACK}, \text{REACH}, \text{TOUCH}, \text{REACH} \triangleleft \text{TOUCH}, \text{TOUCH} \triangleleft \text{REACH}\}$$

$$\mathcal{S} = < p_{st}\ p_{reach}\ p_{touch} >$$

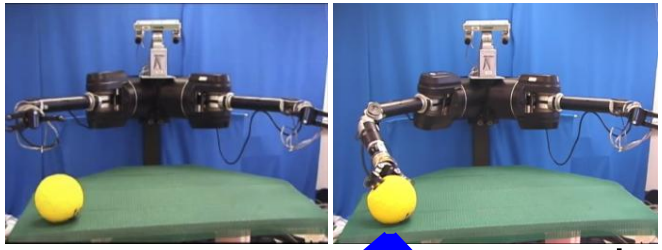
- Achieves a primitive “grab” on simple objects.



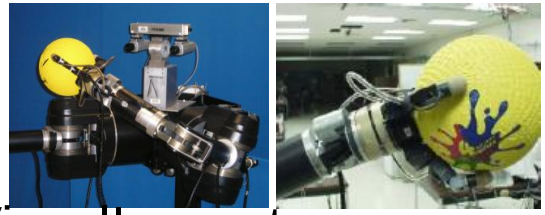


Summary: Skill Development (stages 1-5)

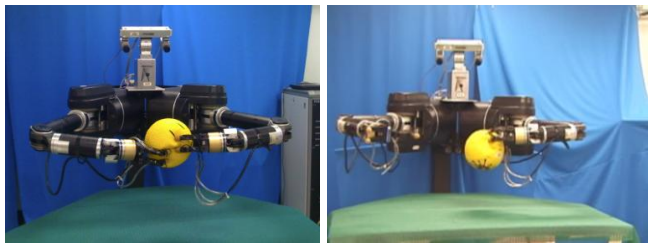
SearchTrack



ReachTouch



VisualInspect

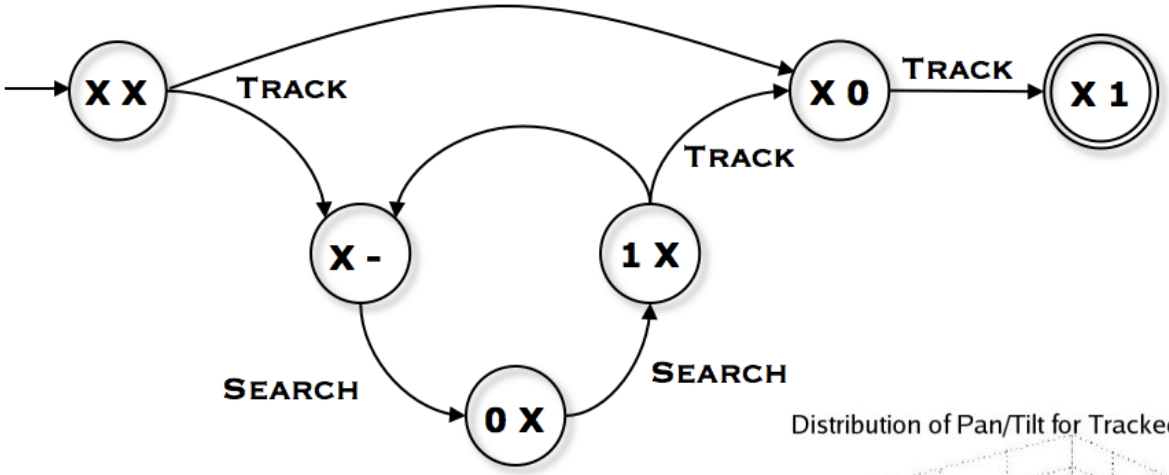


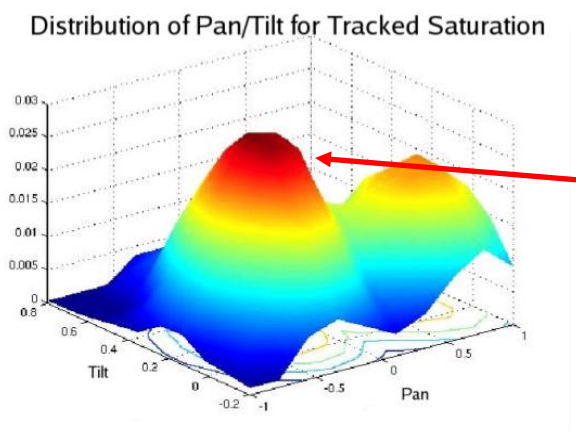
BimanualTouch

PickAndPlace

SearchTrack Generalization

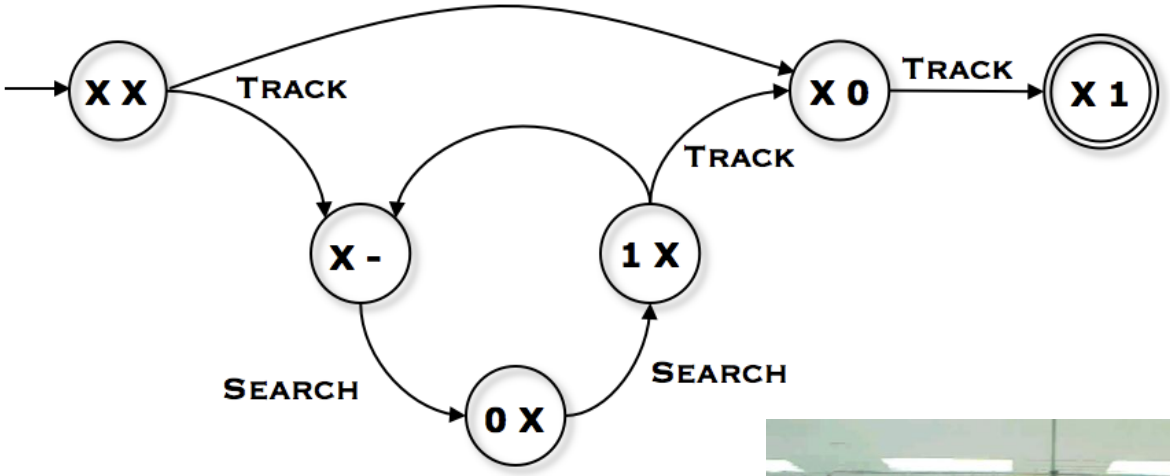
- Learned in the context **highly saturated** regions of pixels.



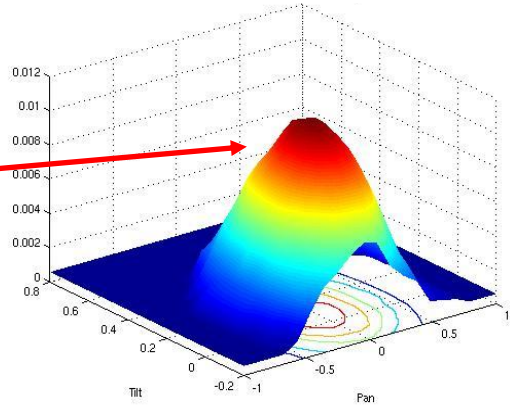


SearchTrack Generalization

- Generalization through the **procedural re-allocation** of sensory signal with **motion** cues.



Distribution of Pan/Tilt for Motion

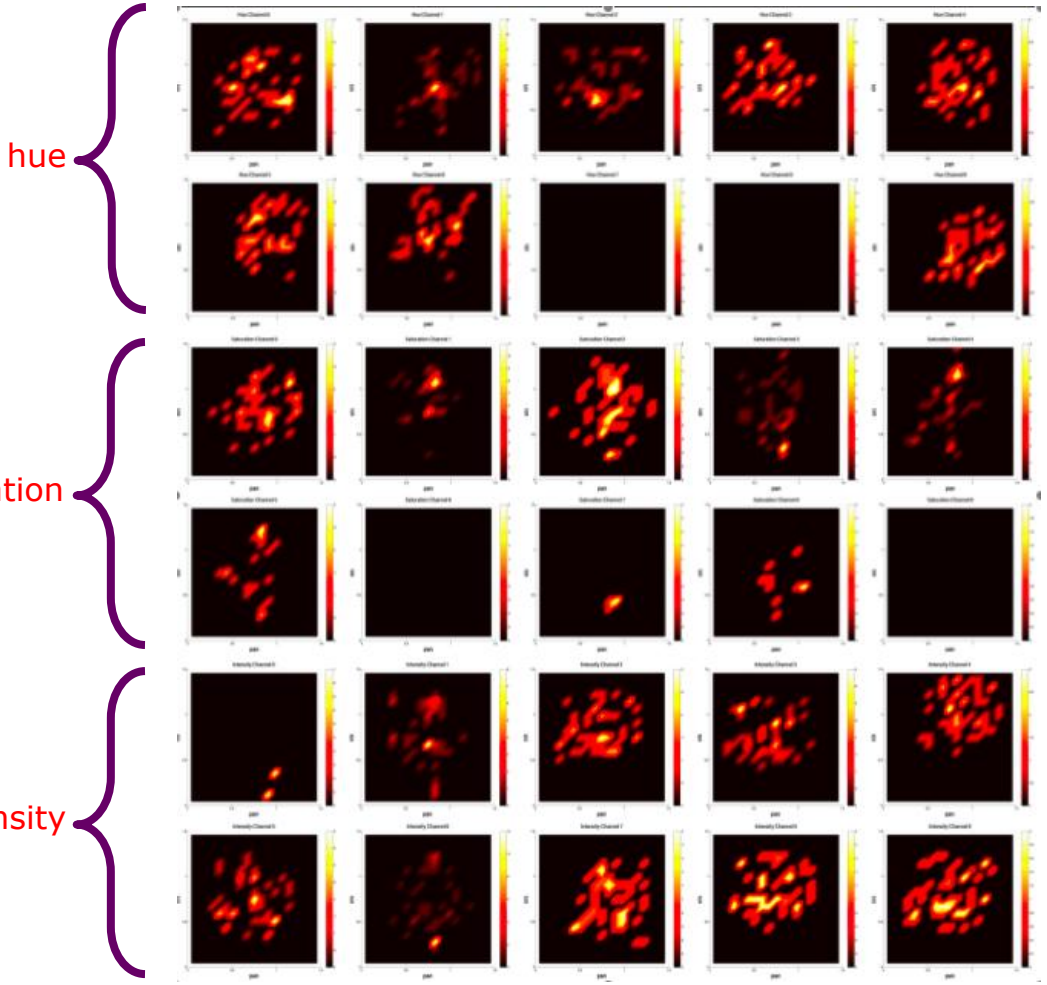


- Same program, *new priors*

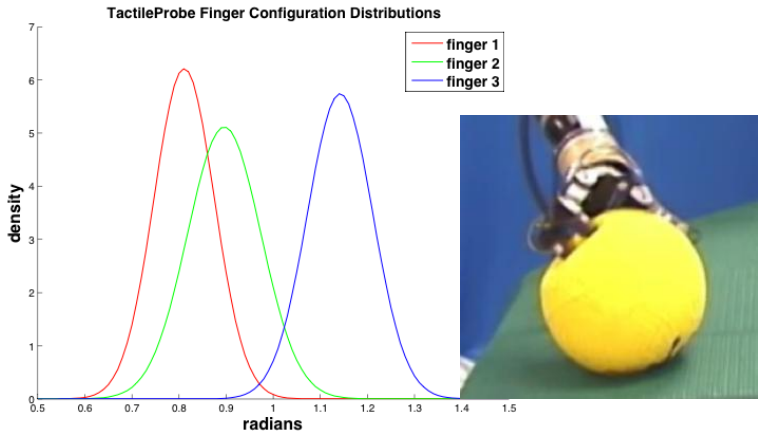
Comprehensive SearchTrack Priors

- pan/tilt models for all 30 channels of visual data:

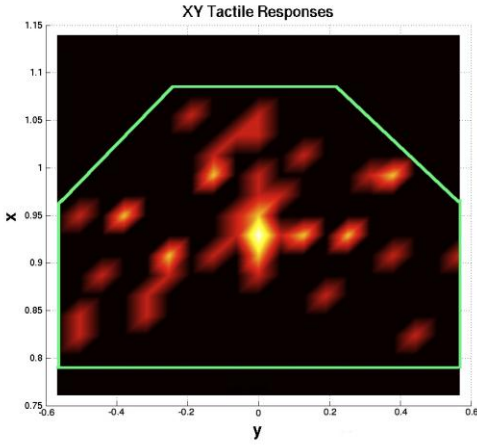
these priors form a primitive visual background model

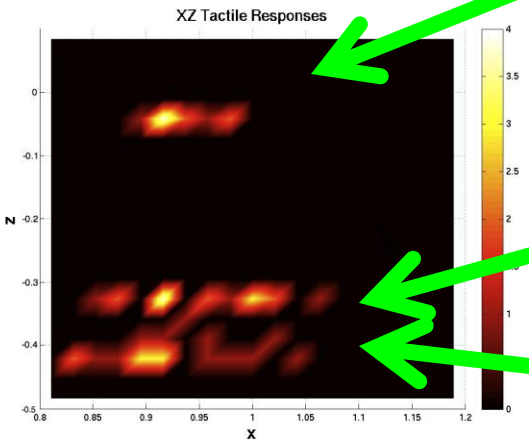


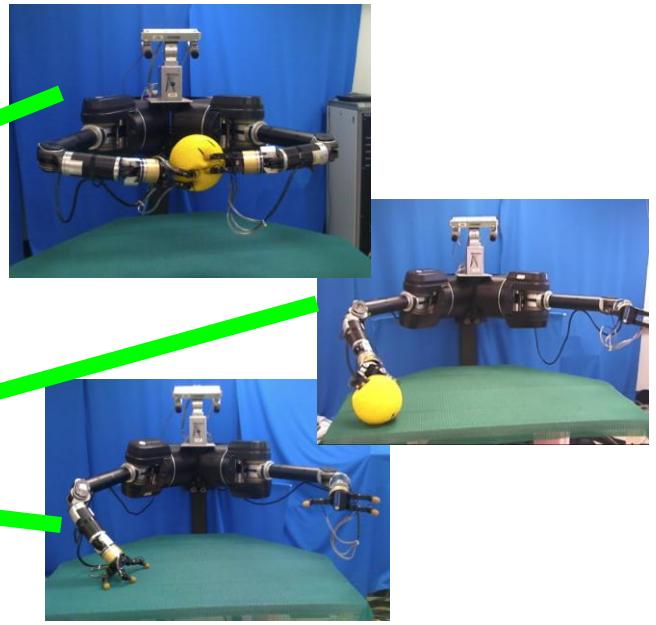
SearchTrack in the Tactile Domain



- Generalized to the **tactile** domain by building Configuration and Cartesian distributions of controlled touch events.







Program Generalization

(Hart et al. - EpiRob 2008)

Program Generalization

For a composite control law

$$c_i = c(\phi_0, \sigma_0, \tau_0) \triangleleft \cdots \triangleleft c(\phi_n, \sigma_n, \tau_n)$$

factor it into **declarative** and **procedural** components

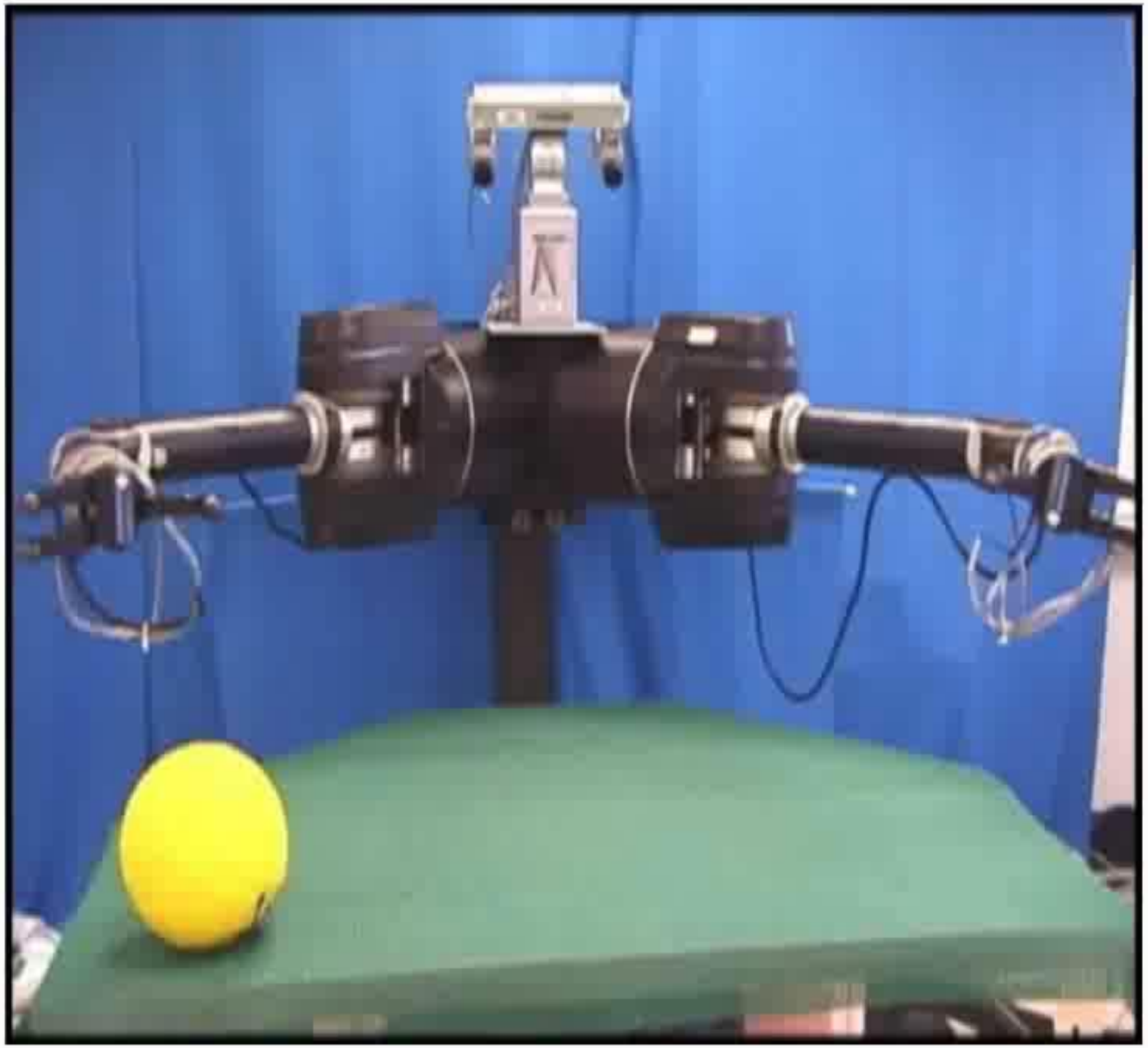
$$\begin{aligned} declarative(c_i) &= (a_0, \cdots, a_n) \\ procedural(c_i) &= (\omega_0, \cdots, \omega_n) \end{aligned}$$

where

$$\begin{aligned} a_m &= a(\phi_m, type(\sigma_m), type(\tau_m)) \\ \omega_m &= \langle \sigma_m, \tau_m \rangle \end{aligned}$$

Learn policy for **procedural allocation** based on context $f \in \mathcal{F}$

$$\psi(a, f) = argmax_{\omega_i} Pr(p_i = 1 | c_i, a, f)$$



Experimental Validation

- Three techniques for learning ReachTouch **locale**, **scale** and **velocity** contingencies.

- **Staged Generalization:**
  - a declarative policy is learned in a *simple context*
  - procedural resource allocations are learned in the more complex context

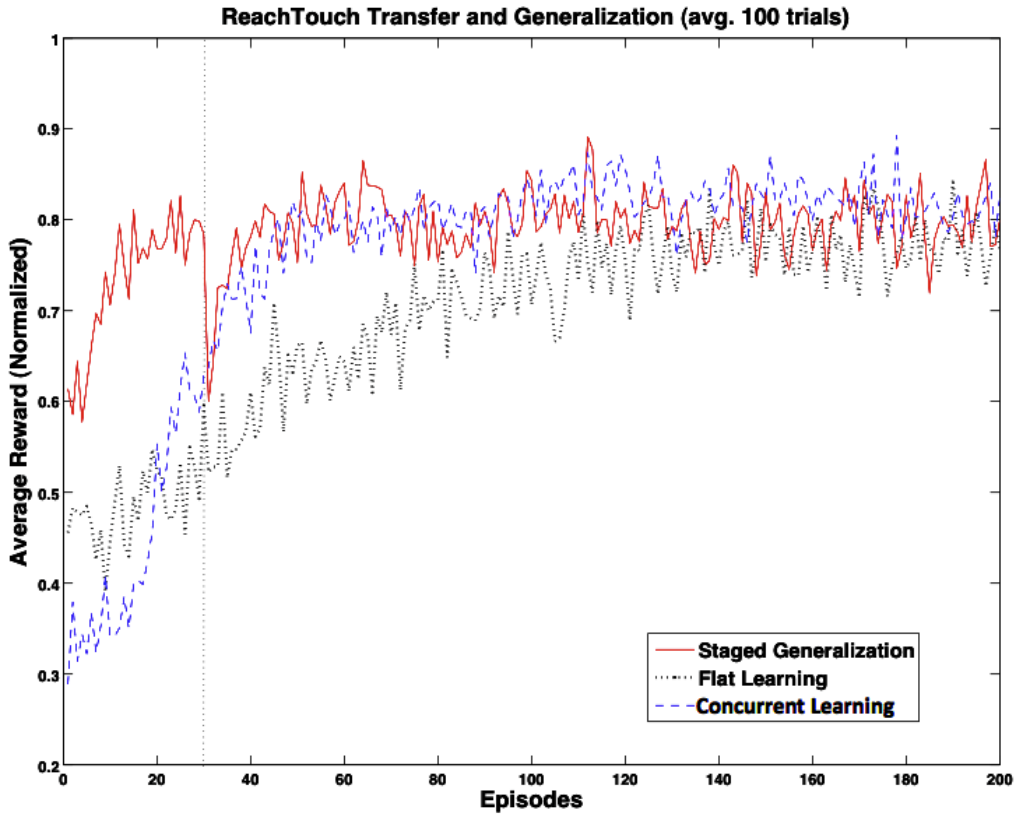
- **Concurrent Learning:**
  - Dexter learns *declarative* and *procedural* policies at the same time in the complex context

- **Flat Learning:**
  - Dexter learns a *single policy* in a *complex environmental context*

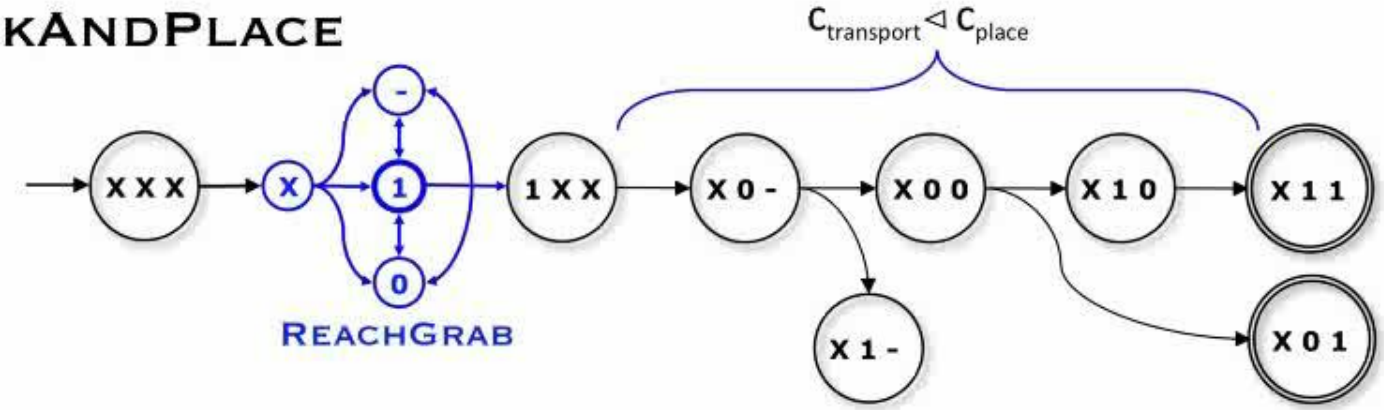
ReachTouch Generalization

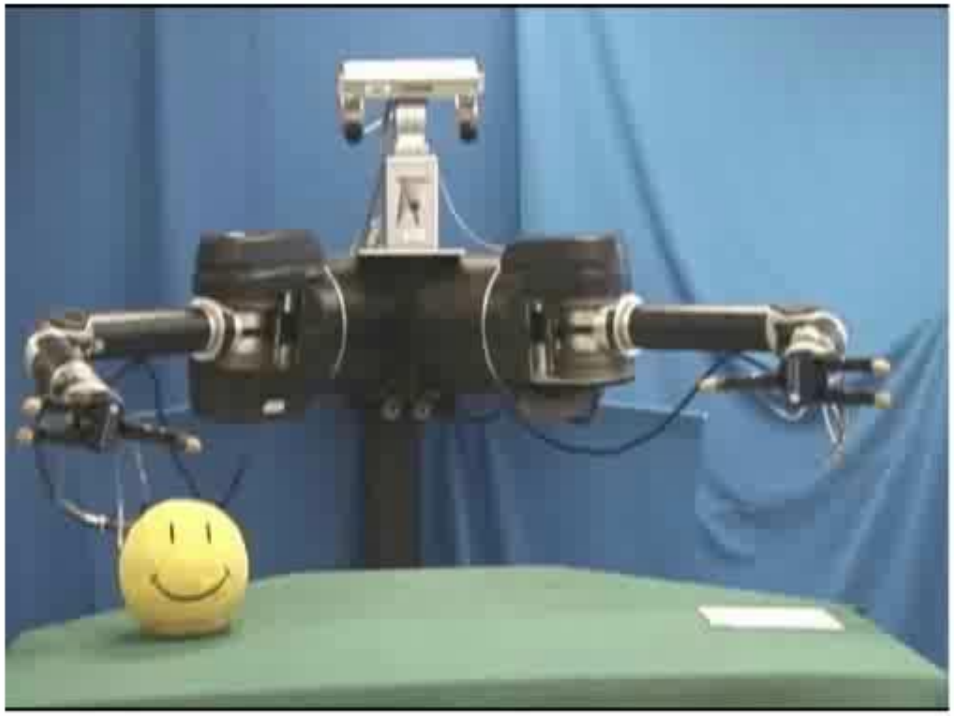
1) declarative learning followed by procedural adaptation
2) Concurrent declarative/procedural learning
3) Flat learner with no generalization

staged generalization has a significant impact on learning performance.



PICKANDPLACE





Affordance Modeling

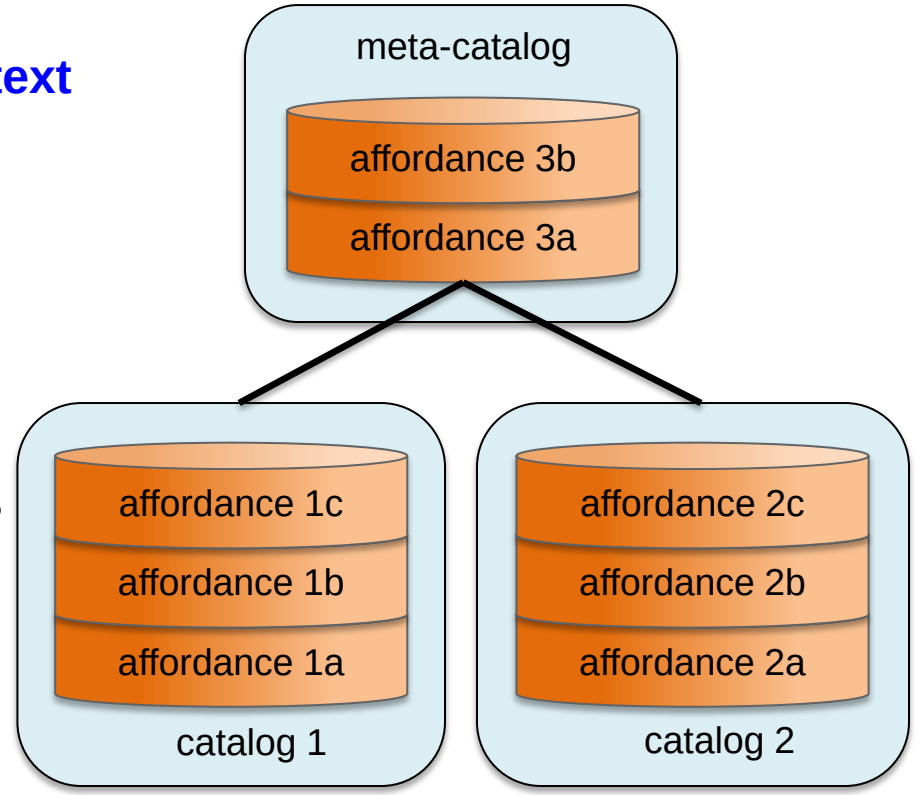
- **Claim:** it will be useful to model the world as a collection of affordances modeled as:

$$P(r|a_i, f)$$

reward action context

Approach:

1. Sample environmental features,
2. Build statistical models of the affordances of those features,
3. Create collections of affordances in structures called **catalogs**,
4. Explore until *habituation*.



(Hart - Ph.D. Thesis 2009)

Habituation Factor

- For model $P(f|a_i, r)$ with variance $\Sigma_i(t)$,
  the **habituation** factor is the change in variance over time:

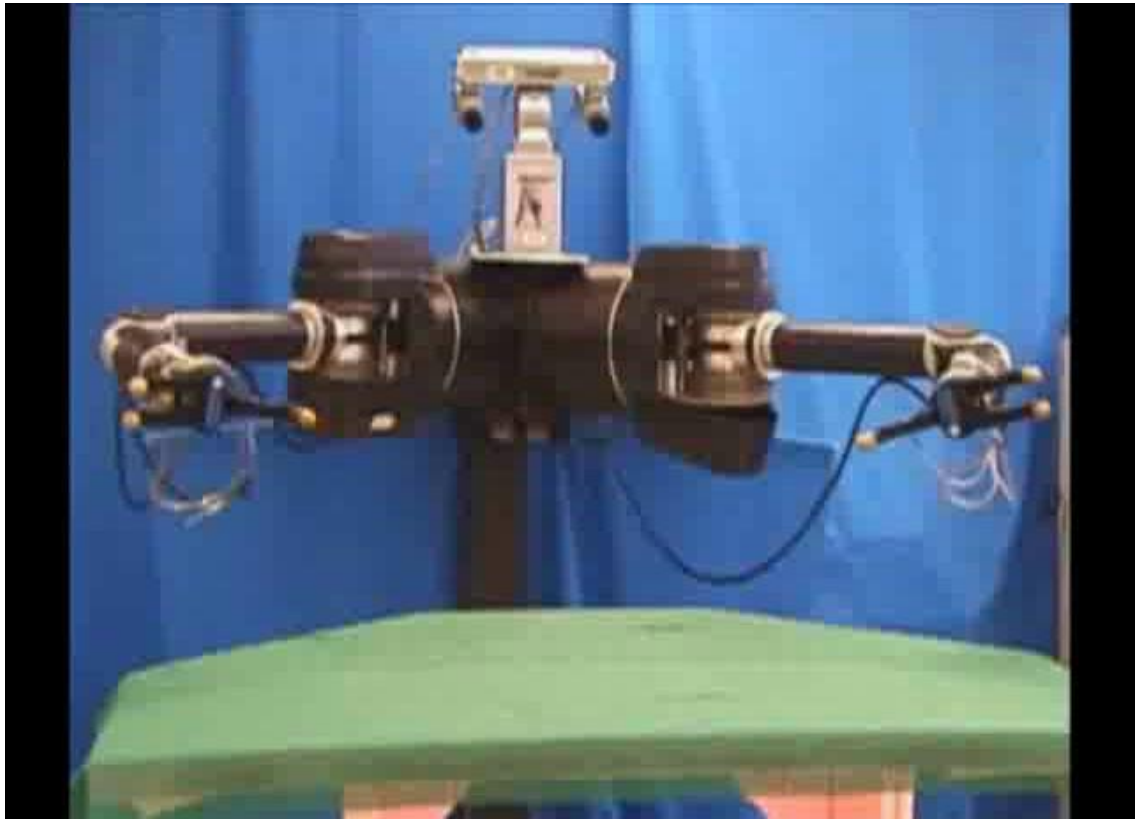
$$h(t) = ||\Sigma_i(t) - \Sigma_i(t-1)||$$

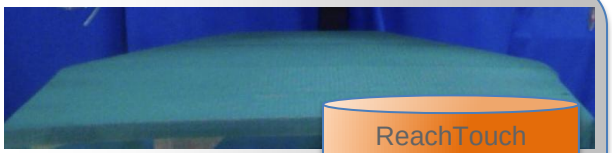
- Modulate the affordance discovery reward based on experience:

$$m = ((p_{k-1} \neq 1) \wedge (p_k = 1))$$
$$r = \left(m \wedge \left(\sigma \subseteq \Omega_{\sigma(env)}\right)\right)$$

Dexter's First Three Catalogs

- Dexter explores three objects until habituation.





GREEN TABLE

ReachTouch

SearchTrack

SMALL ORANGE BASKETBALL

BimanualTouch

ReachTouch

SearchTrack

RED BALL W/ ADDITIONAL HUE FEATURES

$VisualInspect_2$

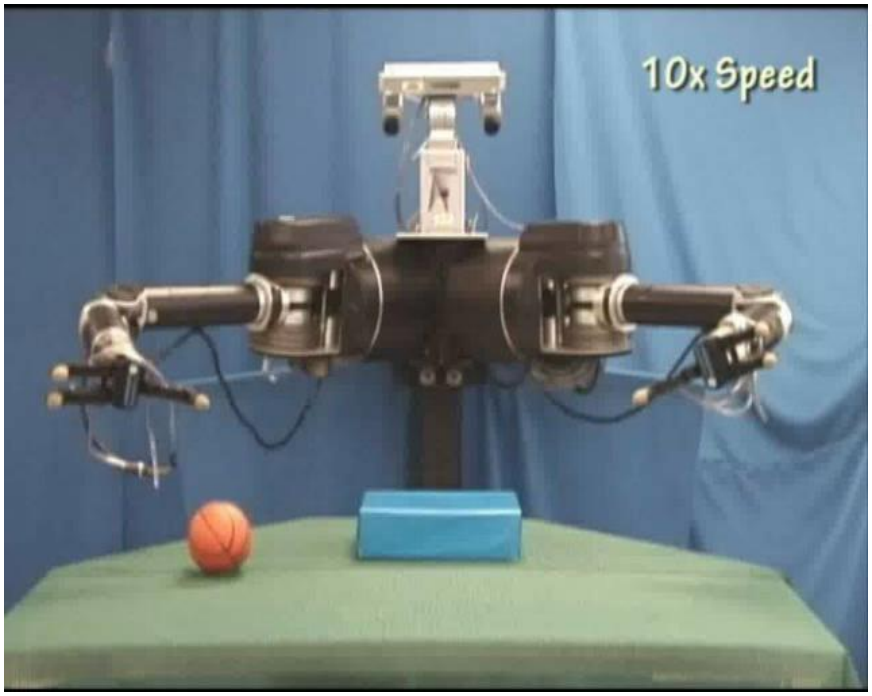
$VisualInspect_1$

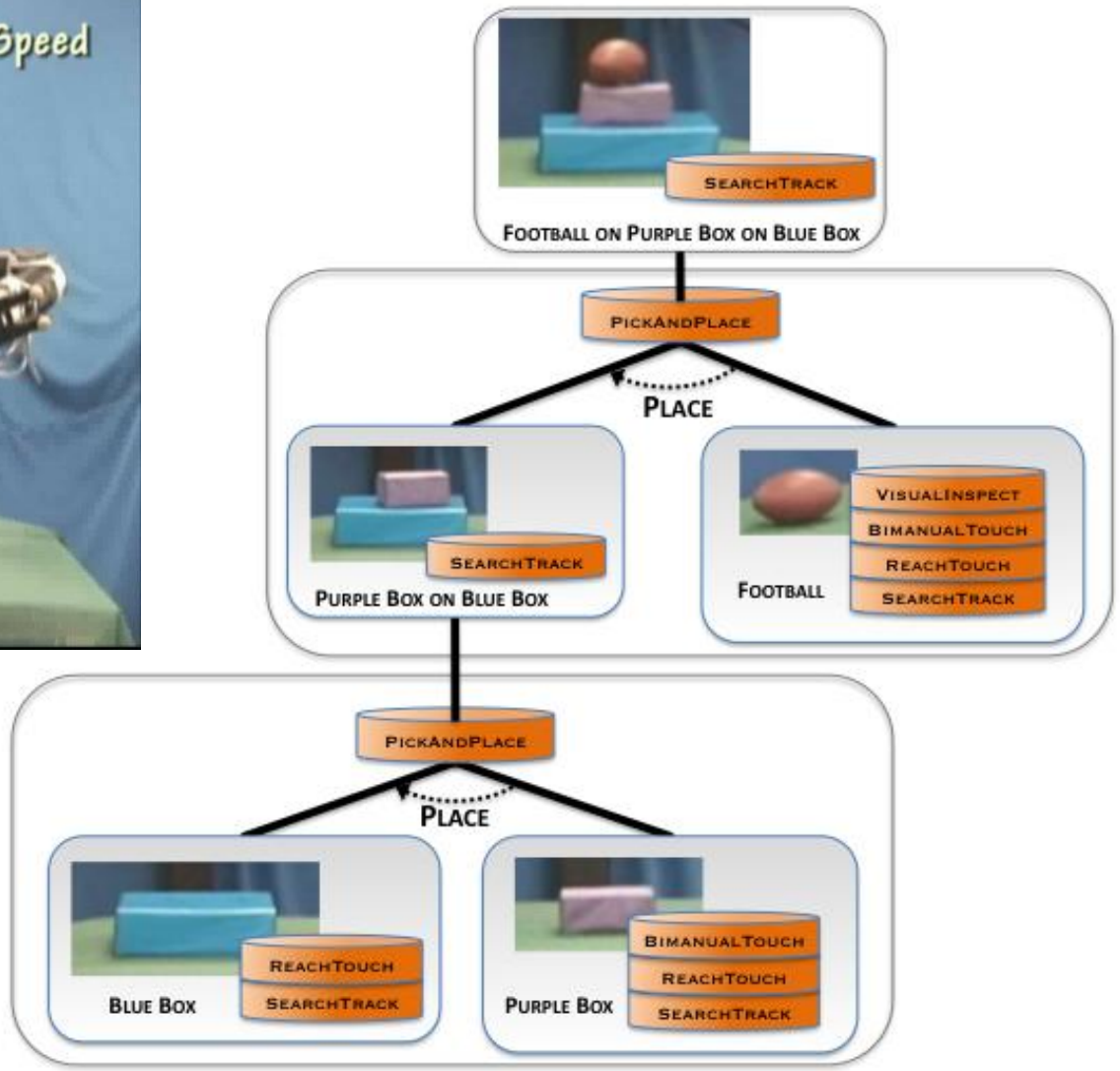
BimanualTouch

ReachTouch

SearchTrack

Exploring Meta-Catalogs

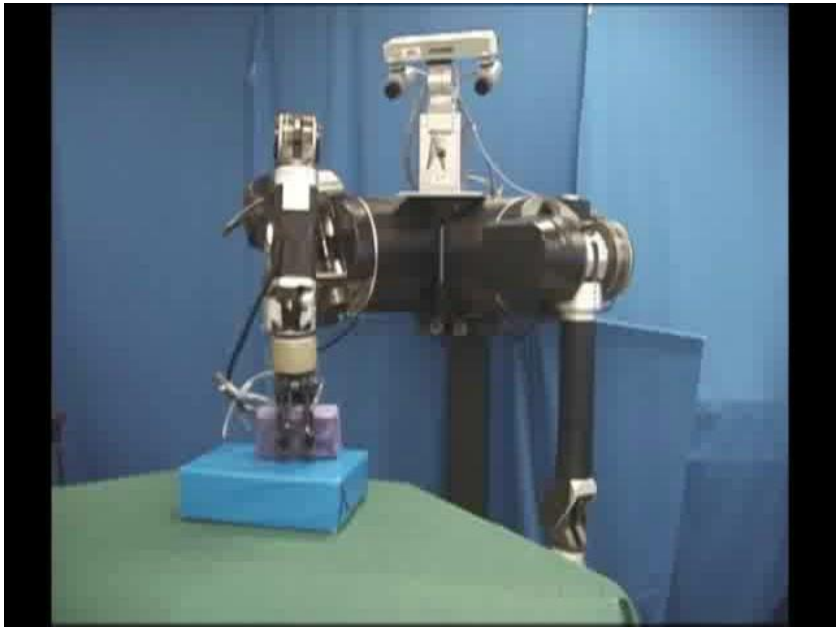


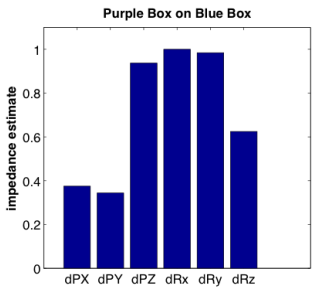


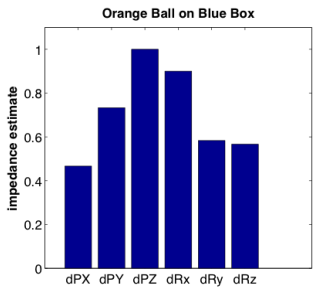
- Dexter explores 2- and 3-way “stacks” using PickAndPlace

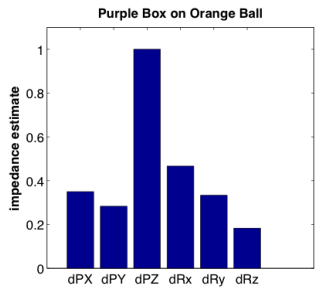
Assembly-Based Affordances

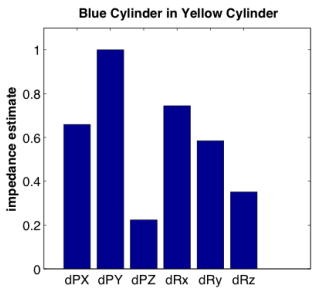
- Multi-feature affordances tested when two visual features come into **contact** with each other.

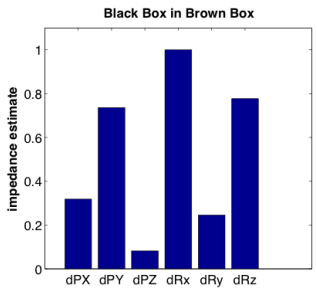


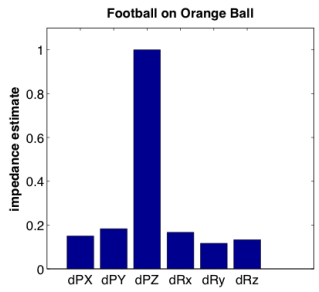










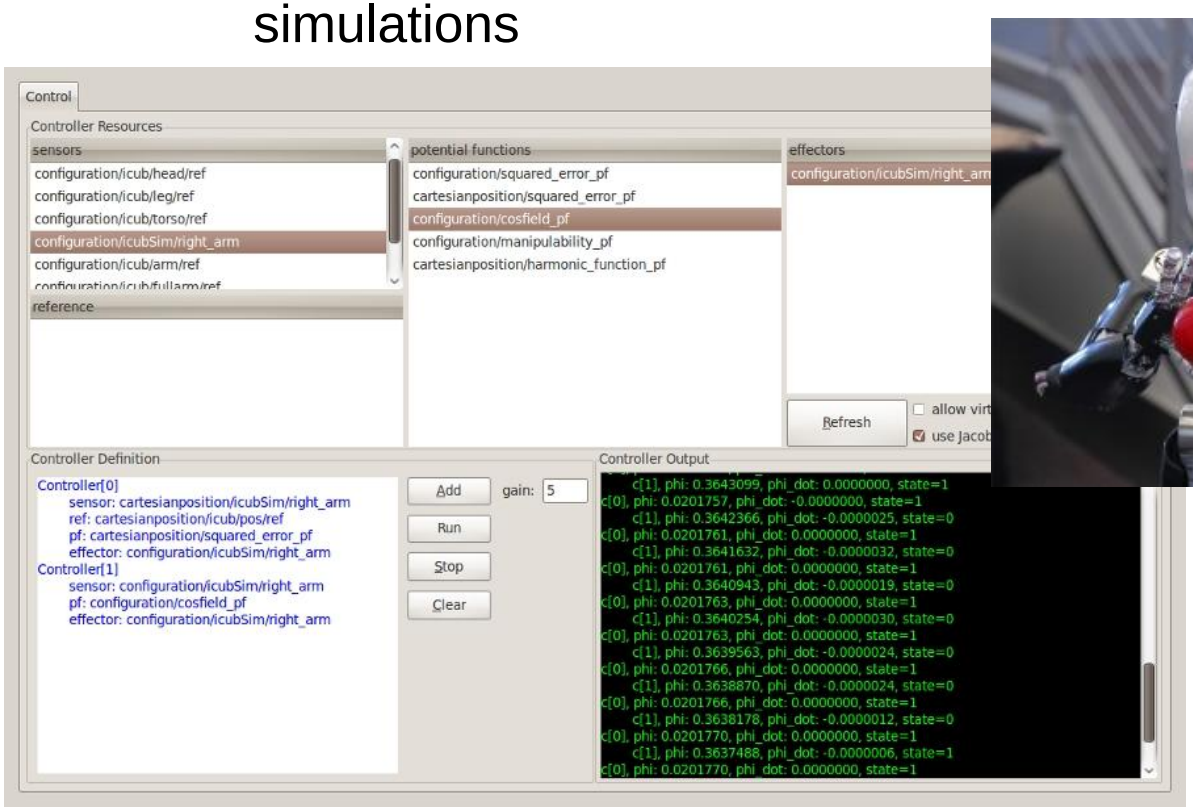


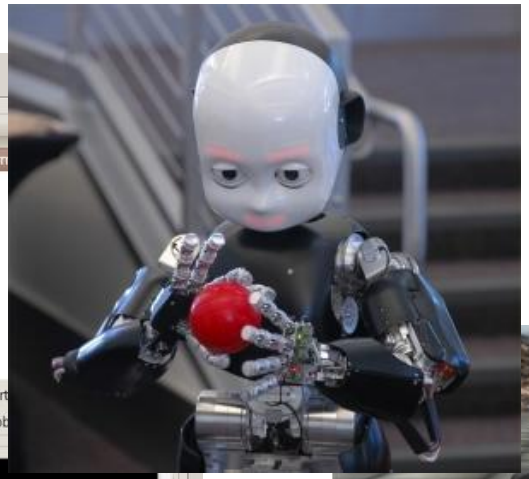
Summary

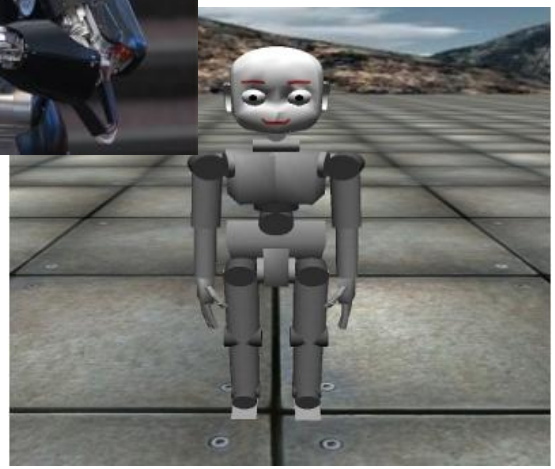
- Actions are constructed through a *general* means of assembling sensory and motor resources into behavioral strategies.
- Knowledge arises from an an inherent desire to find and model control affordances.
- Programs are generalized to provide *dexterous* contingency plans in new contexts.
- These mechanisms form the basis for life-long learning in a robot.

Software API

- The **Control Basis API**:
  - easy transfer of programs between different robots and robot simulations







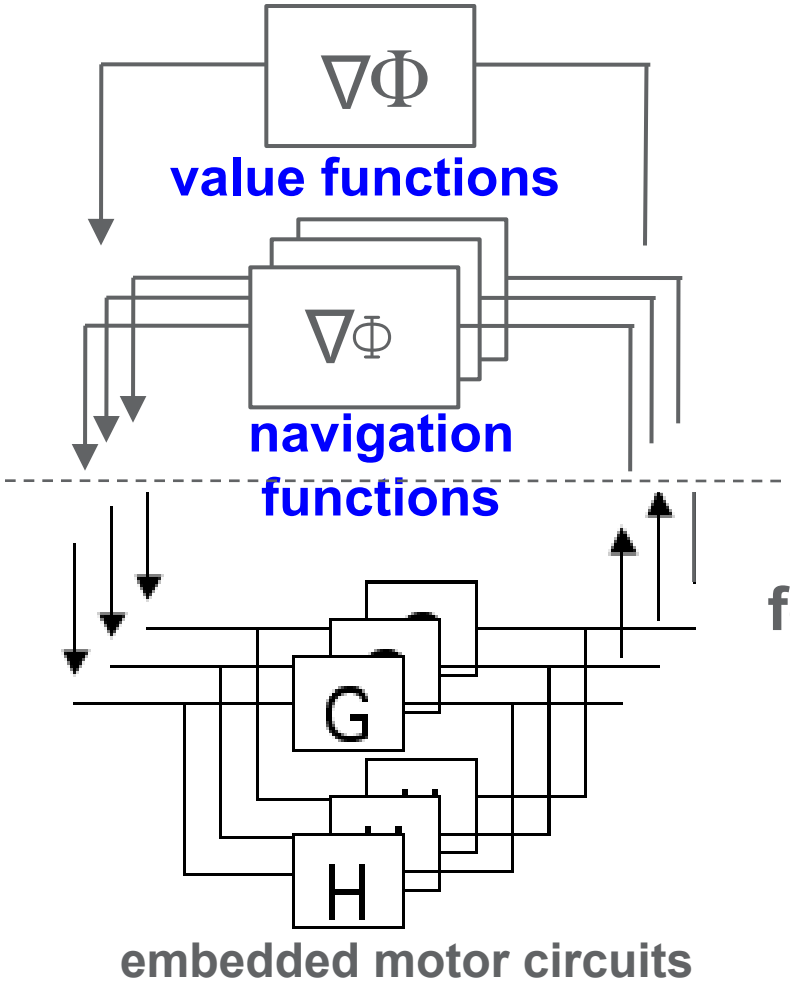
http://sourceforge.net/projects/robotcub/

The End

- Thanks to:
  - Giorgio Metta
  - Rod Grupen
  - Shiraj Sen

The Control Basis

hierarchy of sensorimotor programs



force, position references (τ)

feedback signals (σ)